The Structural Diagnostic Toolset (TSDT)

Instruction Manual

Engine + Modules I–III

Dean Tyldesley

1. Purpose of This Manual

This manual provides complete operational guidance for applying the Structural Diagnostic Toolset (TSDT). It integrates the core engine and all three perimeter modules into a unified framework for practical use.

TSDT is a deterministic structural classification system. It evaluates structural condition under constraint, cumulative irreversible information loss, and finite resolution capacity.

TSDT does not:
- Predict behaviour
- Forecast events or timelines
- Recommend actions
- Optimise solutions
- Evaluate desirability or morality

2. Structural Architecture

TSDT consists of:

• Core Engine - Unified Stress Functional
• Module I - Agency Saturation Calculator
• Module II - Coherence Stress Calculator
• Module III - Invariant Calculator

All modules compute divergence D(x) and apply the unified stress formula:

$$Stress(x) = (D(x) + K) / \rho$$

Where:
D(x) - Module-specific divergence
K - Cumulative irreversible information loss (monotonic)
ρ - Finite resolution capacity

3. Engine Logic and Classification Boundaries

Structural classification is determined by the stress boundary:

Stress < 1 → Stable within structural limits
Stress = 1 → Saturation boundary
Stress > 1 → Structural collapse

Invariant violation supersedes stress and results in:
IMPOSSIBLE (IDENTITY LOSS)

4. General Application Procedure

Step 1: Define the system state space (X).
Step 2: Specify constraints (C) and invariants (I).
Step 3: Select the appropriate module.
Step 4: Compute or bound divergence D(x).
Step 5: Estimate cumulative irreversibility K.
Step 6: Determine resolution capacity ρ.
Step 7: Compute Stress(x).
Step 8: Apply module classification mapping.

5. Module I - Agency Saturation Calculator

Purpose:
Evaluates whether a system retains structurally meaningful choice.

Used For:
- Determining whether real alternatives exist
- Detecting forced motion despite continued action
- Identifying agency saturation prior to collapse
- Distinguishing decision-making from inevitability

Useful For:
- System architects
- AI safety researchers
- Policy and governance bodies
- Risk and failure analysts

Scope:
Defines feasibility of agency only. Does not assess freedom, morality, or recommend actions.

6. Module II - Coherence Stress Calculator

Purpose:
Evaluates whether a system can still resolve meaning under constraint.

Used For:
- Detecting interpretive saturation
- Identifying explanatory breakdown
- Distinguishing disagreement from coherence failure
- Evaluating explanatory load under irreversibility

Useful For:
- Organisational analysts
- Policy and institutional designers
- AI alignment researchers
- Systems theorists

Scope:
Defines feasibility of coherence only. Does not evaluate truth or predict outcomes.

7. Module III - Invariant Calculator

Purpose:
Evaluates whether a configuration can persist without identity loss.

Used For:
- Determining structural impossibility
- Identifying necessary reconfiguration
- Establishing feasibility prior to forecasting or planning

Useful For:
- System and software architects
- Policy and regulatory bodies
- Risk and resilience analysts
- Systems theorists

Scope:
Defines structural feasibility boundaries. Does not forecast events or optimise strategies.

8. Recommended Order of Use

When evaluating complex systems:

1. Apply Module III (Invariant) first to determine structural feasibility.
2. Apply Module I (Agency) to determine whether meaningful choice exists.
3. Apply Module II (Coherence) to determine whether interpretive resolution remains possible.

This order prevents interpretive or strategic analysis of structurally impossible configurations.

9. Structural Limitations

TSDT assumes:

- Constrained systems
- Monotonic irreversibility
- Finite resolution capacity
- Identifiable parameters

Systems outside these assumptions are undefined within TSDT.

10. Version Integrity

This manual corresponds to TSDT v1.0 (Engine + Modules I–III). All applications should reference the same version set to preserve consistency.

Copyright Notice © [Dean Tyldesley], [2025-2026]. This work is proprietary under the Relative Framework. It may be shared and cited with proper attribution; however, the logic, mathematical models, and architectural designs contained herein may not be used commercially or for the development of derivative systems without express written permission.